297. In addition, as reflect in Exhibit L, the remaining asserted claims of the '338 patent are invalid due to obviousness.

298. The *Feather Thesis* related to the field of network monitoring, which was undergoing significant expansion in the 1992 timeframe in which the thesis was written. By spring of 1993, a standard had been developed for doing hierarchical monitoring. Furthermore, the *Feather Thesis* explicitly directed the reader to use SNMP: "[w]ith SNMP now available on many platforms, it would be very easy to do anomaly detection on data from other network components."[139] It would have been obvious to combine the work of the *Feather Thesis* with hierarchical monitoring systems, because the entire network monitoring field was already moving in this direction. In addition, the network monitoring SNMP standards for the purpose of network management documented a wide range of network traffic data categories to monitor. It would have been obvious to combine the *Feather Thesis* with these additional known categories of network traffic in order to improve and expand upon its network monitoring capabilities.

299. In addition, to the extent there were any differences between the configuration of the system described in the *Feather Thesis* and the configuration described in the patents-in-suit, it would have been obvious to modify the configuration of the *Feather Thesis* based upon the nature of the problem to be solved. Such configuration changes would have been motivated by a desire to address the problem of detecting and responding to suspicious network activity in very large enterprise networks.

9. HP Open View and RFCs

300. This section covers the field of network management and in particular the HP OpenView network management platform and the Internet Engineering Task Force (IETF) standards such as SNMP, MIB, and the Remote Network Monitoring (RMON)

[139] *Feather Thesis* at 172.

specifications (RMON I and II) (collectively the "RFCs")[140]. The RFCs define an infrastructure and monitoring capability that can monitor a wide range of measurements for many different network elements, determine if those measurements are unusual, and if so, send alerts to higher-level monitors. The SNMP architecture supported a hierarchical design. Furthermore, network management platforms such as HP OpenView provided a wide range of responses for unexpected behavior, including alerting a user through graphical means, audible means, and sending messages to their pagers.

301. Network management applications such as HP OpenView and the RFCs were not classic intrusion detection systems, but they certainly performed "network monitoring" as claimed in the patents-in-suit. Furthermore, the computer security field is simply a part of the overall network management field.[141] Many network faults and problems create suspicious behavior patterns that are important for intrusion detection systems to be aware of. Many different intrusion detection systems explicitly relied upon network management systems for part of their functionality. For example, NetRanger used HP OpenView as its user interface. NetStalker could send messages to other network management stations via an SNMP trap message. Both the *JiNao Report*[142] and *Emerald 1997*[143] explicitly mentioned using SNMP traffic as a data source for their intrusion detection monitors. In addition, all of the MIB and RMON metrics are designed

---

[140] RFC 1157, A Simple Network Management Protocol (SNMP), May 1990 [SYM_P_0501113- SYM_P_0501142] and [SYM_P_0527111]; RFC 1155, Structure and Identification of Management Information for TCP/IP-based Internets, May 1990 [SYM_P_0501012- SYM_P_0501031]; RFC 1213, Management Information Base for Network Management of TCP/IP-based internets: MIB-II, March 1991 [SYM_P_0501143- SYM_P_0501205]; RFC 1271, Remote Network Monitoring Management Information Base, November 1991 [SYM_P_0501206- SYM_P_0501271]; RFC 2021, Remote Network Monitoring Management Information Base Version 2 using SMIv2 , January 1997 [SYM_P_0603708- SYM_P_0603837].

[141] W. Stallings, SNMP, SNMPv2, SNMPv3 AND RMON 1 AND 2, 3rd ed. 1999.

[142] *JiNao Report* at 5-6.

[143] *Emerald 1997* at 356.

for monitoring network entities such as routers and gateways. Many network attacks such as sweeps and SYN floods are detectable through changes in the standard MIB variables. In fact, almost every network traffic category represented in the patents was represented in the MIBs. Thus, network management applications such as HP OpenView and the RFCs that network management relied upon are directly relevant to the alleged inventions.

302. The network management effort was directed by the Internet Engineering Task Force (IETF), and was worked on by many different organizations and people over the years. The work began in the 1980s, and the first network management documents were published in 1988 (RFC 1067 – SNMPv1 protocol, RFC 1065 – SMIv1 language description, and RFC 1066 – MIB I). Hewlett-Packard developed the HP OpenView network management platform. The HP OpenView for Windows User Guide for Transcend Management Software Version 6.1, which was used for this report, was released in October of 1997 ("*HP OpenView manual*"). However, many other publications document the features of HP OpenView in the relevant timeframe.[144]

303. Network management in general and HP OpenView in particular was designed to help "manage large communications infrastructure."[145] Management included monitoring elements in the network (hosts, routers, bridges, network segments, etc.) for proper or expected behavior, alerting the appropriate entity (a person or another program) when improper or unexpected behavior was detected, and issuing commands to those network elements.

304. The network management paradigm is centered on the concept of a "managed node", a "management application," and a standardized interface. The

---

[144] *See, e.g.*, M. Miller, MANAGING INTERNETWORKS WITH SNMP, 2nd Ed., 1997.

[145] M. Rose, The Simple Book, An Introduction to Internet Management, 2nd Ed., 1994.

management application monitors, and if necessary, sends controlling information to the managed node. Because the protocols (SNMP) and management information base (MIB) description language (SMI) are standardized, a wide range of management applications and managed nodes can be mixed and matched. Furthermore, a management application can in turn be a managed node for a higher-level management application.

305. Salient features of network management systems prior to November 1997 included:

- Developed by a standards body (Internet Engineering Task Force (IETF)).
- The basic network management architecture provided a two-tiered hierarchy of management application and managed nodes.
- A single manager could monitor many managed nodes, and a single managed node could be managed by many network managers.
- The simple two-tiered network management hierarchy could easily be extended to multiple levels of hierarchy, and this was endorsed with the Manager-to-Manager MIB released in 1993.
- The IETF community endorsed a standard network monitoring capability with the RMON and RMON-II specifications.
- The RMON monitors could be run on stand-alone sensors, routers, or on network management platforms.
- RMON MIBs specified a wide range of subjects to monitor, including network segments as a whole, hosts, host-to-host communications, and communication for a particular application (applications are identified by information in the packet).
- RMON MIBs specified a wide range of network metrics to measure for each subject, including number of packets, number of bytes, and many error conditions.
- RMON sensors could be configured to monitor specific metrics and trigger alerts that were sent to higher-level monitors when unusual conditions are detected.
- HP OpenView provided a published API for third-party applications to interact with the OpenView network management application suite.

- HP OpenView provided a number of additional responses including visual alerts, audible alerts, starting additional programs, and sending a message to a pager.

306. Figure 12, which was scanned in from the 1997 book "Understanding SNMP MIBs" by Perkins and McGinnis,[146] shows the basic approach of network management systems. The basic "management station" and "managed node" can be scaled up to support very large networks. At Level 1 is the basic managed node (1). It receives messages from and transmits messages to the Level 2 manager, which in this case is a dual-role entity (2). As described in the text, the "Dual-role entities are typically used to forward SNMP messages (an SNMP proxy), or to consolidate and synthesize information from many systems and make that information available to a higher-level manager" (3). From our security perspective, the "consolidate and synthesize" means to correlate and integrate. The dual-entity object (2) can then forward the "consolidated and synthesized" information to a 3rd tier management station (4), which can also receive information directly from other managed nodes (5).

307. Figure 13 was scanned in from the 1993 book "SNMP, SNMPv2, and CMIP: The Practical Guide to Network-Management Standards" by William Stallings[147] and was originally part of a 1992 article. Of particular interest for this report is remote monitoring, of which the management is specified by the RMON and RMON-II MIBs. Figure 13 shows an example of an RMON deployment for remote monitoring. In the lower left is a network router supporting an RMON monitor (1). At the bottom middle (2) and the bottom right (3) are PCs running RMON monitors. Each of these monitors must report to a higher-level network management station like the one at the top of the

---

[146] D. Perkins and E. McGinnis, UNDERSTANDING SNMP MIBS, 1997. (Level 1, 2, and 3 labels and the circled numbers have been added for this discussion).

[147] W. Stallings, SNMP, SNMPv2 and CMIP, The Practical Guide to Network-Management Standards, 1993.

figure (5). The top-level network management station is also running its own RMON sensor (5). Finally, on the far left is yet another RMON monitor (4), but this one not only reports to (5) but it also supports its own local management console.

308. Figure 14 captures several relevant aspects of RMON sensors. In the upper left corner is a sample of several of the subjects an RMON sensor monitored (1). Furthermore, the RMON MIBs describe a wide range of statistics that may be monitored for the various subjects, including the number of packets, bytes, broadcast packets, error messages, and protocol distributions (2).

309. Figure 14 also briefly shows RMON's approach to anomaly detection and alerting (3). The graph's vertical axis represents the number of packets for some subject (e.g., a host) (4). The RMON MIBs (and indeed, the general purpose MIB-II) is highly configurable as to what metrics are monitored for what subjects. The graph's horizontal axis represents time (5). For this measurement and subject, a normal range of values is determined (6), and an upper threshold (7) and a lower threshold (8) bound that normal range. The RMON sensor then monitors the measurement over time, and as long as the measurements remain in the normal range nothing happens. However, when the measurement first exceeds one of the thresholds of normal values (10), the RMON monitor automatically triggers an alert (11) that is sent to a higher-level monitor.

310. For the RMON sensors, the data source is network packets. For example, it can passively monitor all the packets on a network segment. Most network management agents can monitor additional details about the objects they monitor; for example, the standard MIB specification includes details about the state of a particular connection (i.e., a management application can check if a connection is in a half-open connection). Shown in Exhibit X is a chart of the different types of information that could be monitored via SNMP or RMON sensors.

311. Management applications receive messages from managed nodes or from other management stations. In the case of HP OpenView, individual management applications can be started by and receive messages from OpenView.

312. Any individual variable on a managed device can be polled to determine its current value, and the result of that value can be passed to management applications (more then likely, the management application requested that the value be polled). If the managed node observes a specific value for certain variables, it automatically alerts a network management station.

313. Beginning with the Manager-to-Manager Management Information Base (RFC 1451), the normal range of values for a variable can be defined and if the variable exceeds the normal range an alert is automatically sent to hierarchically higher monitors. HP OpenView extends this by providing support for detecting anomalous behavior for categorical variables too.

314. The network management infrastructure provides a wide range of responses to anomalous or suspicious behavior. At the simplest level, simple events (e.g., a network segment failing) can trigger an alert (via an SNMP trap) to be sent to higher-level monitors. Network management objects can also respond to more sophisticated behaviors by setting alarms to be triggered when a value exceeds a normal range.

315. Furthermore, network management applications are expected to be full-blown applications that can provide complex analysis and response. For example, HP OpenView provided a graphical user interface (GUI) that could automatically alert a user, sound an audible alarm, automatically execute additional programs, or even activate a remote paging device. Finally, network management systems that serve in a dual-role capacity could alert a hierarchically higher network management system. For example,

Perkins says dual-role entities are used to "consolidate and synthesize information from many systems and make that information available to a higher-level manager."[148]

316. As noted in Exhibit M, the *HP OpenView manual* specifically referenced and relied upon the noted RFCs. Furthermore, network management applications such as HP OpenView were publicly used with all of the noted RFCs. In the alternative, given the explicit direction in the *HP OpenView manual* to use the RFCs for network management, such a combination would have been obvious.

317. In my opinion, as more fully reflected in Exhibit M to my Report, the system disclosed in *HP OpenView manual* satisfied every limitation of the indicated claims of the '203 and '615 patents, and enabled one of ordinary skill in the art prior to November 9, 1997 to practice the alleged inventions claimed in these patents. Consequently, the asserted claims of the '203 and '615 patents (except '615 claims 7 and 64-73) were not novel when filed.

318. Furthermore, network management applications such as HP OpenView were in public use with the RFCs for network management prior to November 9, 1997, and this public use also anticipates the indicated claims.

319. In addition, as reflect in Exhibit M, it would have been obvious to combine the statistical detection method from the *Feather Thesis* with the HP OpenView system and Internet Standards, for the reasons discussed previously regarding the *Feather Thesis*, and thus claim 7 of the '615 patent is invalid due to obviousness. Given the fact that HP OpenView managed network infrastructure, and a firewall is network infrastructure, it would have been obvious to treat a firewall as a managed node. Thus, '615 claim 64-73 are invalid due to obviousness.

---

[148] Understanding SNP MIPs, p. 7.

320. In addition, to the extent there were any differences between the configuration of HP OpenView system and the configuration described in the patents-in-suit, it would have been obvious to modify the configuration of HP OpenView based upon the nature of the problem to be solved. Such configuration changes would have been motivated by a desire to address the problem of detecting and responding to suspicious network activity in very large enterprise networks.

10. NetStalker

321. This section covers the commercial product NetStalker sold by Haystack Labs, Inc., as described in the May 1996 document "NetStalker Installation and User's Guide: Version 1.0.2" ("*NetStalker manual*"). Haystack Labs, Inc. developed NetStalker, and the primary developers were Steve Smaha, Steve Snapp, Jessica Winslow, Richard Letsinger, Crosby Marks, Charisse Castagnoli, Brita Womack, and Kristin Johnson.

322. NetStalker enhanced the Network Systems Corp. (NSC) routers by providing richer signature analysis to look for a wider range of misuse than is possible through the NSC router rules. NetStalker also could alert users through a number of means and could automatically modify router filter rules to block detected attacks. Examples in the manual of misuse detected by NetStalker included illegal FTP sessions, SATAN scans, IP spoofing, ICMP error messages, and fragmented TCP headers.

323. NetStalker was a software package that the user installed on a workstation. The NetStalker software was configured to monitor data from and control one or more NSC routers. The NSC routers needed to be configured to send messages to the NetStalker host and accept configuration changes from the NetStalker host. The configuration changes to the NSC routers might include selecting a different set of packets to forward to the NetStaker host or blocking a particular address. NetStalker could also be configured to send messages to other network management stations via an

SNMP trap message. Because the SNMP standards already supported hierarchical architectures, NetStalker systems could be part of a large-scale, hierarchical security management system.

324. Figure 15 shows the basic configuration. At the bottom is an NSC router (which can also operate in bridge mode) (1). The NSC router observed packets, applied a set of filters to each packet, and, depending on the result of those filters, sent a selection of those packets to the NetStalker host (2).

325. The NetStalker system applied a series of filters, where each filter implemented Haystack's pattern matching signature analysis engine (3). If an event passed through all the filters, it reaches the alarm element of NetStalker (4). The alarm element applied simple correlation and a threshold to determine when to take an action such as alerting a user (5) or sending a trap message (6) to a hierarchically higher network management system (7).

326. The original source of data for NetStalker was packets. The NSC router would read the packets (acting as a router or a bridge), select some subgroups of those packets, and forward those packets to the NetStalker host. From those packets, NetStalker could extract numerous metrics from the packets including protocol types (e.g., IP, ICMP, TCP), services (e.g., Telnet, FTP), a wide range of ICMP error messages (e.g., destination unreachable, source quench), and network addresses.

327. NetStalker provided a wide range of responses including: popping up an alarm window on the GUI, sending an SNMP trap message to a network management station, emailing a user, modifying the NSC routers to block an attacker's IP address, and dialing a user's pager.

328. As noted in Exhibit N, NetStalker generated snmp traps, and HP OpenView was designed to use such SNMP traps. Furthermore, both systems were on-sale prior to November 9, 1997. Therefore, the use of the two together constitutes a public use. In the alternative, given that the NetStalker product generated snmp traps and

the *NetStalker manual* stated "[f]or each alarm generated by NetStalker, you can configure one or more alarm handlers to serve as communications channels from NetStalker to you, to other network management tools, or to respond to the alarm"[149] such a combination would have been obvious.

329. In my opinion, as more fully reflected in Exhibit N to my report, the NetStalker product in use with HP OpenView satisfied every limitation of the indicated claims of the '203 and '615 patents, and enabled one of ordinary skill in the art prior to November 9, 1997 to practice the alleged inventions claimed in these patents. Consequently, the claims of the '203 and '615 patents were not novel when filed.

330. In addition, as reflect in Exhibit N, it would have been obvious to combine the NetStalker system with the HP OpenView system and RFCs, for the reasons discussed previously, and thus the indicated claims are invalid as obvious. '615 claim 7 is similarly obvious for the reasons discussed previously with regard to HP OpenView.

331. In addition, to the extent there were any differences between the configuration of NetStalker and the configuration described in the patents-in-suit, it would have been obvious to modify the configuration of NetStalker based upon the nature of the problem to be solved. Such configuration changes would have been motivated by a desire to address the problem of detecting and responding to suspicious network activity in very large enterprise networks.

11. NetRanger

332. I have reviewed in detail the NetRanger documentation listed on Exhibit O. In addition, I spoke to Mr. Daniel Teal in order to confirm my understanding of the functionality of NetRanger. I also reviewed Mr. Teal's expert report and agree with his opinions. I adopt as part of my own report Mr. Teal's description of the NetRanger

[149] *NetStalker manual* at 4-2.

product as well as the rest of the analysis in his expert report, including the Figures and the NetRanger invalidity chart.

333. Under Symantec's alternative claim construction of "monitor" the NetRanger system is still anticipatory. The NetRanger NSX was software that could be reconfigured, for example, to add additional signatures. The NetRanger Director could be reconfigured to consolidate multiple alarms into a single alarm based upon user preferences. The NSXs collected, analyzed and responded to suspicious network activity – using analysis engines and functionality to respond to the detection of events by sending alarms. The NetRanger Director's SAP (Security Analysis Package) correlated events based upon its SQL queries. The NetRanger Director also provided response by alerting the user about problems.

334. In my opinion, as more fully reflected in Exhibit O to my report, the NetRanger product and the NetRanger User's Guide Version 1.3.1 satisfied every limitation of the indicated claims of the '203 and '615 patents (except '615 claim 7), and enabled one of ordinary skill in the art prior to November 9, 1997 to practice the alleged inventions claimed in these patents. Consequently, the indicated claims of the '203 and '615 patents were not novel when filed.

335. I also agree with Mr. Teal's opinion that it would have been obvious to combine NetRanger with a statistical intrusion detection system if one's primary purpose was a research system. Thus, as reflected in Exhibit O, '615 claim 7 and the '212 claims are invalid as obvious.

336. In addition, to the extent there were any differences between the configuration of NetRanger and the configuration described in the patents-in-suit, it would have been obvious to modify the configuration of NetRanger based upon the nature of the problem to be solved. Such configuration changes would have been

motivated by a desire to address the problem of detecting and responding to suspicious network activity in very large enterprise networks.

12. ISS RealSecure

337. RealSecure, as described in the various documents listed in Exhibit P, was a network-based intrusion detection and audit system. It used a signature-based engine to detect a wide range of attacks (e.g., SYN floods), and could generate detailed network audit logs of activities such as which web pages were requested, what users logged into FTP servers, and what files they transferred. The manual recommended placing RealSecure at the organization's gateway to the Internet, either behind the organization's firewall or Internet router. A single RealSecure management console could configure and receive reports from multiple RealSecure sensors spread across the organization.

338. In my opinion, as more fully reflected in Exhibit P to my report, the RealSecure product satisfied every limitation of the indicated claims of the '203 and '615 patents, and enabled one of ordinary skill in the art prior to November 9, 1997 to practice the alleged inventions claimed in these patents. Consequently, the indicated claims of the '203 and '615 patents were not novel when filed.

339. In addition, as reflect in Exhibit P, the remaining asserted claims of the '203 and '615 patents are invalid due to obviousness. To the extent there were any differences between the configuration of RealSecure and the configuration described in the patents-in-suit, it would have been obvious to modify the configuration of RealSecure based upon the nature of the problem to be solved. Such configuration changes would have been motivated by a desire to address the problem of detecting and responding to suspicious network activity in very large enterprise networks.

13. Network Level Intrusion Detection

340. This section covers the University of New Mexico's Department of Computer Science effort to develop a network-based intrusion detection system. The work is described in an August 1990 paper published under the title "The Architecture of a Network Level Intrusion Detection System" ("*NLID 1990*"). The authors were primarily interested in detecting and reacting to computer worms. While aware of misuse detection techniques, the authors wanted to focus on anomaly detection. The metrics their system collected included timestamps (which also provided time of day information), packet length, packet drops, source-destination pairs, and protocol type. These basic metrics could be combined into a wide range of other metrics (e.g., histograms of packet size distributions), and the authors were concerned with determining which metrics were good for intrusion detection and which metrics were not. To perform anomaly detection the authors used a generic algorithm classifier. The classifier would develop a rule base that represented normal behavior so that anomalous behavior could be flagged. The rule base of historical behavior would be periodically updated.

341. In my opinion, as more fully reflected in Exhibit Q to my report, *NLID 1990* satisfied every limitation of the indicated claims of the '338 patent, and enabled one of ordinary skill in the art prior to November 9, 1997 to practice the alleged inventions claimed in these patents. Consequently, the indicated claims were not novel when filed.

342. In addition, as indicated in Exhibit Q, the remaining indicated claims of the '338 patent are invalid as obvious. To the extent there were any differences between the configuration of *NLID 1990* and the configuration described in the patents-in-suit, it would have been obvious to modify the configuration of *NLID 1990* based upon the nature of the problem to be solved. Such configuration changes would have been motivated by a desire to address the problem of detecting and responding to suspicious network activity in very large enterprise networks.

14. '750 Thompson patent

343. This section describes U.S. Pat. No. 5,825,750 "Method and Apparatus for Maintaining Security in a Packetized Data Communications Network," filed by Horace Thompson on March 29, 1996 (the '750 patent). This patent covers packet-based networks; the embodiment of the invention described in the paper is an ATM-based network. However, one of ordinary skill in the art would have understood from the disclosure that the methods described could also be applied to a TCP/IP network given that TCP/IP commonly runs over ATM networks. The '750 patent includes applying both statistical and rule-based analysis to network packets in order to detect possible security problems. For the anomaly analysis, profiles were generated and periodically updated. Packet metrics that were measured included bandwidth measurements and error conditions. Once a potential security problem had been detected, the sensor intrusion detection system could impose itself in the middle of the communication (man in the middle) to protect the secure node, request additional information, or prevent the potential attacker from making a connection with a target system.

344. In my opinion, as more fully reflected in Exhibit R to my report, the '750 patent satisfied every limitation of the indicated claims of the '338 patent, and enabled one of ordinary skill in the art prior to November 9, 1997 to practice the alleged inventions claimed in these patents. Consequently, the indicated claims were not novel when filed.

345. In addition, as indicated in Exhibit R, the remaining claims of the '338 patent are invalid as obvious. To the extent there were any differences between the configuration of the '750 patent and the configuration described in the patents-in-suit, it would have been obvious to modify the configuration of the '750 patent based upon the nature of the problem to be solved. Such configuration changes would have been

motivated by a desire to address the problem of detecting and responding to suspicious network activity in very large enterprise networks.

15. Stake Out

346. This section covers the commercial product called "Stake Out" sold by Harris Corporation. This information is taken from their "Stake Out Network Surveillance White Paper." Stake Out was a network-based intrusion detection that ran on a Sun Sparc 20 computer running Solaris 2.5. It would sniff network packets and process TCP/IP networks using a combination of anomaly detection and misuse detection techniques to detect suspicious behavior. Some of the metrics measured by the system included number of packets, type of packets, and source-destination traffic patterns. Once potentially suspicious behavior was detected, the system would respond by alerting the user, modifying the sensor to collect additional information, and potentially send an alert to a hierarchically higher monitor via an SNMP trap message.

347. In my opinion, as more fully reflected in Exhibit S to my report, Stake Out the product and the printed publication white paper satisfied every limitation of the indicated claims of the '338 patent, and enabled one of ordinary skill in the art prior to November 9, 1997 to practice the alleged inventions claimed in these patents. Consequently, the indicated claims were not novel when filed.

348. In addition, as indicated in Exhibit S, the remaining claims of the '338 patent are invalid as obvious. To the extent there were any differences between the configuration of Stake Out and the configuration described in the patents-in-suit, it would have been obvious to modify the configuration of Stake Out based upon the nature of the problem to be solved. Such configuration changes would have been motivated by a desire to address the problem of detecting and responding to suspicious network activity in very large enterprise networks.

16. Automated Alarm System

349. This section covers Los Alamos National Laboratory (LANL) Automated Information System, as described in: W. Hunteman, "Automated Information System - (AIS) Alarm System," Proc. of the 20th National Systems Security Conference (Oct. 1997) ("*AIS*"). The LANL Alarm System was primarily a hierarchical architecture to integrate a potentially wide-range of security related sensors. The core architecture was a 2-tier architecture where individual sensors sent reports to a "Central Assessment" center, which correlated the reports and optionally took a number of responses including: directing the sensors to modify their analysis, send an email message, close connections, and notify hierarchically higher Central Assessment systems. One example sensor the paper mentioned could be a network sniffer that reported specific patterns of misuse or anomalous patterns of behavior. A Central Assessment component could act as a "Sensor for the next higher level Alarm System", creating a hierarchical monitor. The Alarm System would also provide a standard API for all the components to communicate.

350. In my opinion, as more fully reflected in Exhibit T to my report, *AIS* satisfied every limitation of the indicated claims of the '203 and '615 patents, and enabled one of ordinary skill in the art prior to November 9, 1997 to practice the alleged inventions claimed in these patents. Consequently, the indicated claims were not novel when filed.

351. In addition, as indicated in Exhibit T, the remaining claims of the '203 and '615 patents are invalid as obvious. To the extent there were any differences between the configuration of *AIS* and the configuration described in the patents-in-suit, it would have been obvious to modify the configuration of *AIS* based upon the nature of the problem to be solved. Such configuration changes would have been motivated by a desire to address the problem of detecting and responding to suspicious network activity in very large enterprise networks.

XI. OBVIOUSNESS

352. In the previous sections, I have already explained many reasons why one of skill in the art would have been motivated to combine a particular reference or system with another. In addition to those explanations, I will further elaborate some factors that relate to the obviousness of the alleged inventions. In general when addressing obviousness, I have considered the issue of what a person of ordinary skill in the art prior to November 9, 1997 would have done if confronted by the problem of detecting network intrusions if that person had no knowledge of the patents-in-suit. I have further considered the problem of detecting network-related intrusions across an enterprise network, spanning multiple domains, and how that problem would have been confronted by one of skill in the art in the relevant timeframe.

353. It is important to point out that by November 1997, there was a very rich body of prior art in the network monitoring and intrusion detection fields. Many different individuals and research groups had been working on these issues for many years. Thus, one of ordinary skill in the art would have known about many such systems and past work. Given this large body of prior work, it is impossible for me to enumerate all the possible, likely combinations. My report discusses merely a sample of the possibilities.

354. As discussed earlier with regard to *Emerald 1997*, much of the motivation to look at particular categories of network traffic stems directly from the types of attacks that can be launched against networks. As I understand the alleged inventions of the patents-in-suit, a key issue is whether one of skill in the art would have been motivated to use the claimed network traffic data categories as an input into a network-based intrusion detection system. In my opinion, the claimed network traffic data categories broadly direct one to look at parts of mainly network packet headers, or to look at the volume of data in a packet. If looking for network-based intrusions, or indeed any type of activity

on a network, one of skill in the art prior to November 1997 would naturally have looked at these parts of a network packet.

355. As I described earlier, a network packet typically consists of a header portion and a data portion. The structure of network packets was well-known and indeed trivial in the computer science field in the 1990s. In addition, monitoring and parsing different portions of network packets to gain relevant information about a particular part of a network packet was similarly common in the computer science field.

356. By 1997, it was well-known that attackers could use either the header portion of a packet or a data portion of a packet to carry out an attack.[150] While attacks like SYN floods, IP scans, port scans, and SATAN scans primarily relied on the header portion of network packets, other attacks such as buffer overflow (also known as buffer-overrun) exploits, password cracking, and worms relied heavily on transporting data or code within the data portion of network packets.

357. Attacks that relied primarily on the header portion of network packets used the various data fields contained in the header to carry out the attack. These fields are ordinarily used by various network protocols to administer the transfer of data via network packets. For example, some header fields—such as the SYN and ACK fields in a TCP header—relate to administering the establishment of network connections. Other header fields—such as the source IP address and destination IP address fields in an IP header—relate to administering the transfer of data from one computer to another over a network. Because well-known attacks used such fields in malicious ways, one of ordinary skill in the art would have been motivated to monitor header fields using both profile-based anomaly detection and signature detection.

---

[150] NetRanger 1.3.1 User's Guide at 4-61 (see, e.g., IP Fragment attack and FTP CWD ~root exploit).

358. Other attacks relied primarily on the data portion of network packets to transfer malicious data or code. As noted previously, it was well-known by the early 1990s that several network-based attacks transported malicious code, data, or commands via the data portion of network packets. Because these attacks placed malicious code, data, or commands in the data portion of network packets, one of ordinary skill in the art would have been motivated to monitor the data portion of network packets using both profile-based anomaly detection and signature detection.

359. As I described earlier, network protocols are typically organized as a layered stack, with each protocol being built on the protocol or protocols directly beneath it. It was well-known by 1997 that attacks could be carried out at any layer of the network protocol stack. For example, while SYN attacks were carried out using transport layer TCP flags relating to connection establishment, many attacks directed at FTP servers were carried out using application layer FTP-specific commands. Because each and every network protocol layer was the subject of attacks, one of ordinary skill in the art would have been motivated to monitor each and every layer using both statistical profile-based anomaly detection and signature detection.[151]

360. To summarize, in order to increase the number of attacks that could be detected, one of ordinary skill in the art would have been motivated to use intrusion detection systems to analyze all portions of a network packet, including the header and data portions of packet formats from each network protocol layer. The nature of the problem to be solved would have led one of skill to examine the claimed network traffic data categories.

---

[151] The need to monitor each and every network protocol layer is related to the need to monitor both the header and data portions of network packets, since packets at a given layer typically encapsulate packets at a higher layer. For example, in order to monitor transport-layer TCP segments, one has to extract those segments from the data portion of network-layer IP packets.

361. By 1997, it was also well-known to those of skill in the art that attackers targeted all types of network entities, including specialized entities like routers, gateways, proxy servers, firewalls, proxy servers, and any other device connected to a network. There were several reasons why attackers targeted a wide variety of network entities. First, because of the prevalence of use of standard network protocols, an attack that exploited one of these standard protocols could typically be launched against the wide array of network entities using that protocol. For example, a SYN flood that was initially developed to attack servers utilizing TCP could just as easily be launched against a handheld networked device that also utilized TCP. Thus, just as network-based intrusion detection leveraged the network protocols to monitor several different types of network entities, attackers could also leverage the use of these protocols to expand their base of attack targets.

362. Second, attackers had an incentive to attack these specialized network entities that comprised the internet infrastructure itself. Infrastructure entities like routers and gateways were attractive targets because an attack on those targets would be felt by every network entity that used the router or gateway as an intermediary for communication to the rest of the network. In other words, attacks on routers and gateways potentially would be more widely felt. Because of the importance of these entities that comprised the network infrastructure itself, one of ordinary skill in the art would have been motivated to protect these entities using intrusion detection systems.

363. By November 1997, there also existed a strong motivation to design intrusion detection systems to interoperate with both other intrusion detection systems and other types of network security devices. Standards-based efforts such as CIDF had emerged to promote interoperability through the use of common protocols and common APIs. There were several factors driving this move towards greater interoperability. From a technical standpoint, greater interoperability enhanced the ability of an intrusion

detection system to analyze information from a wider range of network security devices (like firewalls). Greater interoperability also enhanced scalability by allowing intrusion detection systems to work and communicate with each other regardless of whether these systems were administered by the same party. In this way, cooperation enabled by greater interoperability would allow intrusion detection systems to scale up to larger networks. From a commercial standpoint, greater interoperability allowed new products to work with and leverage existing deployments of heterogeneous network security solutions.

364. Both the SRI IDES/NIDES/EMERALD team and UC Davis' Computer Science Laboratory were well-known players in the intrusion detection space in the 1990s. Both groups published, presented publicly at conferences, and were involved in related DARPA projects. One of ordinary skill in the art would have known about these groups, and would have considered them to be doing related work. Given that both teams were attempting to solve problems relating to network intrusion detection, it would have been obvious to one of skill in the art that it would be worthwhile to combine systems from these two groups.

365. In addition, given the number of "surveys" and compilations discussing various different intrusion detection systems in existence prior to November 9, 1997, there was ample direction and motivation to combine such systems. One such example[152] is a 1994 paper on which I am a named author: B. Mukherjee, L. Todd Heberlein and K. N. Levitt, "Network Intrusion Detection," IEEE Network May/June 1994 ("*NID 1994*").[153] This paper notes that "[t]he intrusion detection problem is becoming a

[152] For additional examples in 1997, see L. Todd Heberlein, Network Radar presentation, 24 July 1997; and http://web.archive.org/web/19971011083618/www.hokie.bs1.prc.com/ia/N2-TODD.htm

[153] I understand that a copy of this publication was actually produced from the files of Mr. Porras, one of the named inventors. *See* SRI 058251.

challenging task due to the proliferation of heterogeneous computer networks."[154] The paper also points out that it is common to combine statistical and rule-based systems, and provides the reader with several examples of each: "[t]ypically, IDSs employ statistical anomaly and rule-based misuse models in order to detect intrusions…" "[i]n this paper, several host-based and network-based IDSs are surveyed, and the characteristics of the corresponding systems are identified."[155] The paper encourages further investigation: "new and more-effective detection strategies must be investigated."[156] The paper specifically encourages investigation into intrusion detection for large networks: "much more research is expected to be conducted, e.g., how can the intrusion-detection concept be extended to arbitrarily large networks…"[157]

366. Given these specific pointers in *NID 1994* to a variety of existing intrusion detection systems, as well as the specific direction to combine statistical and rule-based approaches and expand IDS to better cover arbitrarily large networks, one of skill in the art would have been motivated to combine the systems discussed in this paper, as well as other related systems, to achieve the goals of the paper. Attached in Exhibit V is a chart demonstrating how the references and explanations of different existing systems in *NID 1994* anticipate many of the asserted claims of the patents-in-suit. In the alternative, the *NID 1994* paper makes it obvious to combine all these systems. As further indicated in Exhibit V, the remaining asserted clams are rendered obvious in combination with the indicated references. It would have been obvious to combine the disclosure in *NID 1994* with these additional systems for the reasons discussed previously. *NID 1994* encouraged one of skill to investigate and combine existing systems.

---

[154] *NID 1994* at 26.
[155] *NID 1994* at 26.
[156] *NID 1994* at 41.
[157] *NID 1994* at 41.

A. SECONDARY CONSIDERATIONS

367. I have been informed that it is appropriate in analyzing the obviousness of an alleged invention to consider "secondary considerations" of non-obviousness. I understand that secondary considerations include: commercial success of the invention; satisfying a long-felt need; failure of others to find a solution to the problem; and copying of the invention by others. Other secondary considerations include licensing by competitors and contemporaneous recognition of the inventor's achievements.

368. Based upon my review and understanding of the facts, as well as my own personal knowledge, I am not aware of any secondary considerations supporting a finding that the patents-in-suit were not obvious.

369. In my opinion, the Emerald system disclosed in the patents-in-suit did not satisfy any long-felt need that I was aware of. I was very familiar with intrusion detection systems in the late 1990s and was deeply involved in the field. I do not recall anyone identifying any "need" fulfilled only by the Emerald system. My peers in the intrusion detection field were aware of the Emerald system through conferences, but I do not recall any particularly unique feature of Emerald that was considered extremely valuable to the intrusion detection community.

370. Furthermore, I am not aware of anyone copying the Emerald system.

371. Many other systems, some of which have been discussed in this report, were able to successfully detect computer intrusions and in particular network attacks. For example, the NSM system and its many different incarnations, such as ASIM, were successful in detecting network attacks. (NSM became ASIM, which as indicated in Mr. Teal's report was widely used by the Air Force). Thus the government, which funded Emerald, was actually using other intrusion detection systems, including some that started out as research-oriented projects such as NSM. In addition, DIDS, another

research-oriented system from UC Davis, was very successful in developing useful correlation features to track users across different systems, solving a key problem in computer security.

372. It is important to recognize that there is a difference between actual commercial products and research funded by the government. Laudatory statements regarding research projects do not necessarily translate into success in the commercial world, because research projects are evaluated on a different set of metrics than actual commercial projects. Research systems typically are run in a limited, lab environment, whereas commercial systems need to run robustly in a messy, real-world environment. Furthermore, research projects typically only address a small part of the problem, whereas commercial systems need to have a rich supporting infrastructure to make them usable products.

373. In the commercial realm, the NetRanger system was a commercial success used by Fortune 500 companies to protect their networks from intrusions. NetRanger was also successful in government testing, with the DOD/SPOCK report stating:

> "Results of the tests clearly demonstrated that when properly configured, the NetRanger hardware/software package:
>
> 1) Can be used to detect, report, and act on intrusion related activities launched across a network with a high degree of accuracy,
>
> 2) Would detect all attempted penetrations signatures contained in the default list as installed in the NetRanger for this demonstration,
>
> 3) Can be used to provide practical and effective intrusion detection, reporting, and selected automatic response actions." [158]

[158] DOD/SPOCK Report at 2 [SYM_P_0074255- SYM_P_0074481 at SYM_P_0074263].

In addition, the DOD/SPOCK report concluded "[i]n the true sense, this suite of tests proved the viability of Real-Time Network Intrusion Detection and Response for implementation today, in a warfighter networked environment."[159]

374. As noted in Mr. Teal's report, the government also actually purchased and used the NetRanger product. I do not believe that the EMERALD system ever achieved the success that NSM/ASIM and NetRanger did.

XII. INVENTORS' FAILURE TO DISCLOSE BEST MODE

A. Legal standard

375. I understand that the patent laws require that if an inventor knows of a best mode of practicing the claimed invention, the inventor must disclose that best mode. I also understand that the legal standard for this best mode requirement involves two factual inquires: (1) a subjective determination of whether the inventor had a best mode of practicing the claimed invention; and (2) if the inventor had a best mode of practicing the claimed invention, an objective determination of whether the best mode was disclosed in sufficient detail to allow one skilled in the art to practice it. I have been asked to render an opinion regarding both of these inquires.

B. Documents relied upon and methodology

376. As part of my inquiry into whether the inventors satisfied the best mode requirement, I have reviewed many documents. For the purposes of this litigation, SRI has placed into escrow a computer containing source code[160] related to the patents-in-suit.

---

[159] DOD/SPOCK Report at 5.4 [SYM_P_0074255- SYM_P_0074481 at SYM_P_0074287].

[160] In this report, I am using the term "source code" to encompass not only the text files that are compiled or interpreted in order to generate an executable software component, but also any associated initialization files, configuration files, or other types of input files that are read in by the software component.

Although the sheer number of source code files made it impossible for me to analyze every single source code file on the computer, I have performed numerous electronic searches across the entire escrowed computer and also extensively browsed the file system of that computer. Based on this searching and browsing, I identified many relevant files and extensively reviewed those files. I have also reviewed various documents produced by SRI in this litigation, various SRI publications, the patents-in-suit, the *Live Traffic Analysis* and *Statistical Methods* articles incorporated by reference into the patents-in-suit, and the appendix of source code filed with the patents-in-suit.

377. In addition, in order to understand the inventors' subjective beliefs as to their best mode of practicing the alleged inventions at the time of filing, I have reviewed emails from the inventors in the relevant timeframe. I have also reviewed transcripts of the inventors' deposition testimony as well as testimony from others working on the Emerald project at the time (furthermore, I actually attended many of these depositions).

378. In my review of the source files that SRI placed into escrow, I made use of the RCS and CVS software systems that were provided with the escrow computer and by which some of the source code files were maintained.[161] Both RCS and CVS are software systems known as "version control" systems. Such systems are also commonly referred to as "revision control" systems. Version control systems are frequently used in environments where (1) it is important to be able to retain previous versions of files; and (2) several different users are making changes to the same set of files. Version control systems are often used by software development teams. For each version of a file, a version control system typically maintains a copy of that version and information associated with that version, including the date that version was made, who made the

[161] For a description of CVS, *see* "Version Management with CVS," available at http://ximbiot.com/cvs/manual/. For a description of RCS, *see* "Official RCS Homepage," http://www.cs.purdue.edu/homes/trinkle/RCS/.

version, and any comments that the author of the version wrote to describe that version. Earlier versions of files and the information regarding those earlier versions can be retrieved at a later point in time. Both RCS and CVS provide the above functionalities.

379. For the files maintained by RCS or CVS, I used RCS commands or CVS commands respectively to determine the dates associated with various versions of the files. For the files not maintained by RCS or CVS, I used standard Unix commands to view the last modification date of the files.

C. The eXpert signature engine, SRI's best mode for analyzing network traffic data, was not disclosed

380. To determine whether the inventors had a best mode for practicing the claimed inventions, I first reviewed the source code underlying several SRI software components that were developed prior to November 9, 1998. One such software component for which I reviewed the underlying source code was a signature engine developed by SRI named eXpert (I believe this is pronounced "e-expert").[162] Among the source code files underlying eXpert, there were several files containing P-BEST rules. Because the rules in these files acted as signatures for analyzing network traffic data, I will use the term "P-BEST rule" and "P-BEST signature" interchangeably. Before I

[162] I understand that Mr. Porras has testified that the source code file eXpert.c, source code files from the "libunit" library, source code files from the "libmessage" library, and source code files from the "libapi" library are all part of the set of the source code files that form the code base for eXpert. *See* Porras 30(b)(6) Depo. Tr. 193:23-195:19. To the extent that these files were included on the escrowed computer or in the appendix to the patents-in-suit, I reviewed those files. I also reviewed several source code files that contain core functionalities of eXpert but that Mr. Porras failed to mention. Mr. Porras has also testified that a "knowledge base" is also part of eXpert. Porras 30(b)(6) Depo. Tr. 195:2-13. Based on my review of the source code underlying eXpert, I have ascertained that the "knowledge base" that Mr. Porras alluded to in his testimony is a suite of P-BEST rules described later in my report.

explain how the various subcomponents of eXpert work, I will explain briefly the use of expert systems in misuse or signature detection engines.

1. Expert systems

381. An expert system is a computer program that uses a set of rules to derive conclusions. I am familiar with several types of expert systems and at trial I may testify regarding these systems. The rules used by expert systems are often called "inference rules." Inference rules are of the form "if X then Y." An example would be, "If you are eligible to vote, then you are at least 18 years of age." Note that these rules can be much more complex and involve several Boolean operators. An example of a slightly more complex rule would be: "If X and Y, then Z." Some expert systems employ a technique known as "forward chaining." Forward chaining is the process of applying inference rules to facts in order to reach conclusions or "goals."

382. Signature engines are frequently implemented using an expert system. Note that the expert system itself does not typically provide any signatures. Domain specific signatures must be written using the expert system's rule grammar in order for the expert system to be applied to a particular domain.

2. SRI's P-BEST expert system

383. There are many well-known, publicly available expert systems.[163] Since the 1980s, SRI has used an internally developed forward-chaining rule-based expert system called P-BEST.[164] Based on my review of SRI publications, it is my understanding that in 1998 SRI viewed P-BEST as superior to other forward-chaining

---

[163] Perhaps the most well-known, publicly available expert system is CLIPS. *See* http://www.ghg.net/clips/CLIPS.html.

[164] *See* "Detecting Computer and Network Misuse Through the Production-Based Expert System Toolset (P-BEST)," Proceedings of the 1999 IEEE Symposium on Security and Privacy, Oakland, California, May 9-12, 1999 [SYM_P_0068713-28] (hereinafter "P-BEST 1999") at 6-7.

rule-based expert systems, and had consistently used P-BEST in its intrusion detection projects (including IDES, NIDES, and EMERALD).[165] In touting P-BEST in October 1998, Mr. Porras (one of the inventors) and his colleague Mr. Lindqvist pointed to its speed, its easy-to-use grammar for defining rules, and its close integration with the C programming language.[166] P-BEST consisted of both runtime libraries and a translator that translated rules written in the P-BEST rule grammar to C code. This translation allowed the rules to be applied at runtime with increased speed. The speed with which the rules could be applied was an extremely important factor in SRI's choice to use P-BEST to implement eXpert, because the P-BEST signatures would need to be applied to real-time network traffic.

3. SRI had developed a fully-functional version of eXpert prior to November 9, 1998

384. I have analyzed the version of eXpert that SRI had developed as of November 9, 1998. Specifically, I have analyzed the C source code underlying eXpert—including the source code underlying P-BEST—and the suite of P-BEST rules that SRI had developed for eXpert. I have also analyzed the version of estat that SRI had developed as of November 9, 1998 in order to compare eXpert with estat.[167] Based upon my review of the source code, deposition testimony, and communications from the inventors described in this section, it is my opinion that as of November 9, 1998 the inventors considered eXpert/P-BEST to be their best mode of practicing network

---

[165] *See* P-BEST 1999 at 3.

[166] P-BEST 1999 at 1. Although this paper was published in 1999, I have reviewed a draft of this paper dated October 1998 that expresses the same praise for P-BEST expressed in the published version. *See* SRI 151720-40.

[167] As discussed previously in my report, estat is a generic code for implementing statistical profiling. I have called it "generic" because the estat code included in the appendix is not specific to analyzing network traffic, but instead is designed to accept any type of data as long as it is formatted into a structure estat can understand.

surveillance / network monitoring / event monitoring for detecting "suspicious network activity" as claimed in the patents-in-suit.[168] Furthermore, the inventors also considered eXpert/P-Best to be their best mode of practicing "a signature matching detection method" as claimed in the '212 patent.

385. I understand that various SRI witnesses have given conflicting testimony regarding the eXpert signature engine. There has been conflicting testimony regarding the extent to which code for eXpert was included in the appendix to the patents-in-suit.[169] There has also been conflicting testimony regarding the functional state of eXpert as of November 9, 1998.[170] For example, Mr. Porras has testified that he was not sure when SRI first integrated P-BEST with the rest of eXpert.[171] Despite this conflicting testimony, it is clear from my review of relevant documents and source code that as of

---

[168] Several documents indicate that SRI had implemented versions of eXpert by the July 1998. *See* Email from P. Porras to K. Skinner and U. Lindqvist entitled "emerald-ftp delivery" (July 7, 1998) [SRIE 0400354-55] (referencing compiled version of eXpert); Email from P. Neumann to Cathy Schott entitled "Re: FW: FY99 Incremental Funding (September 3, 1998) [SRIE 0020286-88](stating that SRI "had now completed integration of the EMERALD eXpert as well as the EMERALD statistics component"); SRI Source Code REQUEST # 3.

[169] Porras 30(b)(6) Depo. Tr. 212:25-216:11.

[170] Porras 30(b)(6) Depo. Tr. 212:25-216:11.

[171] Porras 30(b)(6) Depo. Tr. 190:5-23, 212:25-216:11. Because Mr. Porras's testimony alone does not provide an adequate basis for me to determine whether he believed at the time of filing that he possessed a best mode for practicing the claim inventions, I have looked to documents produced by SRI in this litigation that are contemporaneous with or prior to the filing of the shared specification underlying the patents in suit. Specifically, I have reviewed an email sent by Mr. Porras to Mr. Lindqvist on November 9, 1998, the exact date of the filing of the application underlying the patents-in-suit. *See* Email from P. Porras to U. Lindqvist entitled "Re: 1999 IEEE Symposium on Security and Privacy Paper Submission (fwd)" (November 9, 1998) [SRIE 0275123]. In this email, Mr. Porras informs Mr. Lindqvist about the work that SRI had performed over the prior two weeks in analyzing test data provided by MIT's Lincoln Labs as part of the DARPA 1998 offline-evaluation of intrusion detection systems. Mr. Porras states that SRI used eXpert to successfully detect several attacks in Lincoln Labs test data while estat "didn't produce any meaningful results." Mr. Porras's assessment of eXpert on November 9, 1998 confirms my determination, described below, that as of November 9, 1998 SRI had implemented a fully functional version of eXpert.

November 9, 1998, SRI had a fully-functional version of eXpert that was integrated with P-BEST.

386. Based on my analysis of the source code placed into escrow by SRI, I have determined that prior to November 9, 1998 SRI had developed a fully-functional version of eXpert. As described below, prior to November 9, 1998, SRI had an implementation of P-BEST, code that integrated P-BEST with the rest of eXpert, and a complete suite of P-BEST signatures to analyze network traffic data.

a. P-BEST inference engine and translator

387. As mentioned earlier, the 1999 paper on P-BEST authored by Mr. Porras and Mr. Lindqvist states that SRI used P-BEST in its earlier IDES and NIDES intrusion detection systems well before 1998. Nevertheless, I reviewed several source code files, listed in Exhibit Y, underlying the P-BEST inference engine and P-BEST translator to confirm whether SRI had a working implementation of P-BEST as of November 9, 1998. My review of the files in Exhibit Y confirmed that as of November 9, 1998, SRI had a fully-functional implementation of the P-BEST software described in the 1999 P-BEST paper.

b. Integration between P-BEST and other parts of eXpert

388. I also analyzed the eXpert-related source code files to confirm that eXpert used P-BEST as its expert system. I have confirmed that as of November 9, 1998 eXpert used P-BEST as its expert system. The files that illustrate this integration are listed in Exhibit Z. It is important to note that while earlier versions of eXpert.c contained the relevant integration code, during the summer of 1998 that integration code was migrated to the source code file event.c.[172]

[172] *See* SYM_P_0549715 – 18.

c. Suite of P-BEST signatures for analyzing network traffic data

389. In order to determine the capabilities of eXpert as of November 9, 1998, I analyzed the P-BEST signatures that SRI had written as of November 9, 1998. The files that I analyzed are listed in Exhibit AA. Taken together, the rules in these files make up an entire suite of signatures for analyzing several types of network traffic data, including but not limited to data transfers, network packet data transfer commands, network packet data transfer errors, network connections, network connection requests, network connection acknowledgements, and ICMP packets that may include error codes.

i. The suite of P-BEST signatures analyzed several different types of network traffic data

390. I have analyzed the set of P-BEST rules contained in the file named ftp.pbest labeled as REQUEST # 22. These files analyze a number of different aspects of FTP-related traffic. FTP is an internet protocol that stands for "File Transfer Protocol." I am familiar with FTP and other application-layer network protocols such as HTTP, NFS, SMTP, and Telnet, and I may testify at trial regarding these and other application-layer network protocols. I have determined that the rules in this file monitor, among other things, data transfers, network packet data transfer commands, and network packet data transfer errors.[173]

391. I have analyzed the set of P-BEST rules contained in the file named syn_attack.pbest and labeled REQUEST # 28. I am familiar with SYN attacks, and I may testify at trial regarding how SYN attacks operate and how they can be detected and

[173] My analysis has been confirmed in part by the testimony of Mr. Lindqvist, who modified parts of eXpert during the summer of 1998. Specifically, Mr. Lindqvist testified that the signatures in Request 22 analyze requests for data transfer. Lindqvist Depo. Tr. 110:3-11.

thwarted. I have determined that the set of rules in syn_attack.pbest analyze, among other things, network connections, network connection requests, and network connection acknowledgements.[174]

392. I have analyzed the set of P-BEST rules contained in the file named rules.pbest and labeled Request # 30. The rules in this file analyzed network traffic data in order to detect an IP sweep. An IP sweep is a method by which to scan a network in order to ascertain the IP addresses of the nodes attached to the network. I am familiar with several types of sweeps and scans, and at trial I may testify as to how various sweeps and scans operate, how they can be detected, and how they can be thwarted. I have determined that the set of rules in Request # 30 analyze, among other things, ICMP packets that may include error codes.[175]

ii. The P-BEST signatures were fully functional

393. I understand that SRI witnesses have characterized this suite of P-BEST rules aimed at analyzing network traffic as "experimental."[176] I disagree. Based on my analysis of each of the rule sets comprising the suite of rules, I have concluded that each rule set would have been successful in detecting suspicious network activity. In fact, virtually identical rules from this suite ultimately appeared in a paper published by Mr. Porras and Mr. Lindqvist in 1999. Drafts of this paper from October 1998 show that by October 1998 Mr. Porras viewed several of these rules as mature and capable of detecting

---

[174] My analysis has been confirmed in part by the testimony of Mr. Lindqvist, who testified that the signatures in the file syn_attack.pbest (REQUEST # 28) analyzed "one type of network connection request" and "one type of network connection acknowledgment or lack thereof." Lindqvist Depo. Tr. 119:16-120:1.

[175] My analysis has been confirmed in part by the testimony of Mr. Lindqvist, who testified that the signatures in the file marked as REQUEST # 30 may analyze ICMP error codes assuming that the ICMP_code field was referenced and assuming that the ICMP_code field corresponds to ICMP error codes. U. Lindqvist Depo. Tr. 137:6-17.

[176] Porras 30(b)(6) Depo. Tr. 190:5-23, 212:25-216:11.

SYN attacks.[177] In October 1998 Mr. Porras believed that the SYN attack eXpert rules were "done."[178] Mr. Porras also wrote, "eXpert is already capable of detecting SYN Floods."[179] Furthermore, SRI used eXpert during an important intrusion detection evaluation effort by their sponsor that occurred in late October and early November 1998.[180]

4. eXpert was SRI's best mode for practicing signature detection upon network traffic data

394. As noted previously, based upon my review of the source code, deposition testimony, and communications from the inventors described in this section, it is my opinion that the inventors also considered eXpert/P-Best to be their best mode of practicing "a signature matching detection method" as claimed in the '212 patent. In fact, as of November 9, 1998, eXpert was the only signature engine among the EMERALD-related components that had been implemented.

5. eXpert was SRI's best mode for analyzing network traffic data

395. I have compared the functionalities of estat and eXpert. It is my opinion that eXpert would have outperformed estat in analyzing network traffic data. Specifically, eXpert would have been able to detect more suspicious network activity than estat while at the same time generating less "false positives" than estat. In the context of intrusion detection systems, a false positive occurs when the system reports an intrusion when in fact there is no intrusion. A problem endemic to profile-based anomaly detection systems like estat is that they ordinarily generate more false positives than signature detection systems. Although I have not tested the estat component, I have

[177] *See* SRI 151720-40.
[178] SRI 277983-8002 at 277986.
[179] SRI 277983-8002 at 277987.
[180] *See* SRIE 0275123; SRI 0010873.

analyzed the statistical algorithms used by estat. Based on my analysis of those algorithms, I have concluded that estat would suffer the same problems regarding generating false positives that other profile-based anomaly detection engines suffer from.

396. In order to determine whether at the time of filing the inventors believed that eXpert was the best mode for analyzing network traffic data, I reviewed a twenty page document written by Mr. Porras in mid-October 1998 entitled, "EMERALD eXpert/estat—TCP/UDP/ICMP Analysis Summary."[181] In this document, Mr. Porras discusses 35 different attacks and analyzes whether estat, eXpert, or both estat and eXpert are well-suited to detect each attack. Exhibit BB summarizes Mr. Porras's October 1998 conclusions. The October 1998 document clearly shows that Mr. Porras believed eXpert to be more effective overall than estat in detecting intrusions. As shown in Exhibit BB, Mr. Porras estimated that estat would detect eight of the 37 different attacks, while eXpert would detect 25 of the 37 different attacks. Furthermore, for several attacks, Mr. Porras wrote that estat would not be effective because "estat probably won't see any significant anomalies here."[182]

6. In the alternative, the combination of eXpert and estat was the inventors' best mode for analyzing network traffic data

397. Although it is my opinion that the inventors believed at the time of filing the application underlying the patents-in-suit that eXpert was the best mode for detecting "suspicious network activity," I have been asked to assume for the sake of argument that the inventors believed that estat was better than eXpert for this purpose. If the inventors believed estat to be better than eXpert, it is my opinion that the inventors would have also viewed a combination of estat/eXpert to be the best mode for detecting suspicious

---

[181] [SRI 277983].

[182] *See, e.g.*, SRI 27798328002 at 277986.

network activity; better than estat alone. In other words, even if the inventors believed that estat was better than eXpert in detecting suspicious network activity, the inventors would have believed that the best mode for detecting suspicious network activity would have been a combination of estat and eXpert. Thus, in my opinion the failure to disclose eXpert would still run afoul of the best mode requirement even if the inventors believed estat to be better at detecting suspicious network activity than eXpert.

7. eXpert was not disclosed in the patents-in-suit

398. Given my determination that eXpert was SRI's best mode for detecting suspicious network activity at the time of the filing of the application underlying the patents-in-suit, the next question in the best mode analysis is whether eXpert was disclosed in sufficient detail to allow one skilled in the art to practice it. In order to answer this question, I analyzed the patents-in-suit, the articles incorporated by reference into the patents-in-suit, and the source code appendix to the patents-in-suit.

399. I have reviewed the source code to appendix to the patents-in-suit and I have attempted to identify the names of each file in the appendix by comparing the files in the appendix to the files on the source code computer that SRI placed into escrow. Because the appendix was not included on the source code computer, it was not possible to do a character-by-character comparison of the files in the appendix and the files on the source code computer. In spite of this technical limitation, I have determined to the best of my ability the names and file paths for each of the files contained in the source code appendix. These file names and file paths are listed in Exhibit CC.

400. The source code files for the most substantial and important subcomponents of eXpert were not included in the source code appendix. First, the appendix contains none of the source code files—listed in Exhibit Y—for either the P-BEST inference engine or P-BEST translator.

401. Second, the appendix does not contain any source code showing the integration between P-BEST and the rest of eXpert. The appendix contains version 2.2 of the source code file eXpert.c.[183] This version of eXpert.c contains none of the code related to the integration between the rest of eXpert and P-BEST. Prior to the summer of 1998, the code that integrated P-BEST with the rest of eXpert was contained in the source code file eXpert.c. However, in the summer of 1998, SRI moved this integration code and much of the functionality of eXpert.c into a source code file called event.c.[184] Versions 2.0 and later of eXpert.c do not contain this integration code. SRI included version 2.2 of eXpert.c in the appendix, but failed to include event.c. The result is that SRI failed to include in the appendix any file that showed how P-BEST was integrated with the rest of eXpert.

402. Third, the appendix contains none of the files—listed in Exhibit AA—containing P-BEST signatures. In fact, the nine hundred plus page source code appendix does not contain a single P-BEST rule.

403. I also analyzed the patents-in-suit and the two articles incorporated by reference[185] into the patents-in-suit to determine whether any of those documents contained information related to P-BEST, the integration of P-BEST with the rest of eXpert, or the suite of P-BEST rules that SRI wrote to detect suspicious network activity. I found no discussion of P-BEST, the integration of P-BEST with the rest of eXpert, or the suite of rules that SRI wrote to analyze network traffic data and detect suspicious activity in any of those documents.

---

[183] *See* [SYM_P_0549715-18].
[184] *See* SRI Source Code REQUESTS #19, #20, #23.
[185] As explained previously in my report, it is my understanding that SRI has admitted that these two articles may not be used for satisfying the best mode requirement. Nevertheless, in an abundance of caution, I analyzed them anyway.

404. To summarize, it is my opinion that the signature engine eXpert was SRI's best mode as of November 9, 1998 for detecting suspicious network activity. It is also my opinion that SRI failed to sufficiently disclose eXpert.

D. etcpgen, SRI's best mode for performing network monitoring, was not disclosed

405. Another SRI software component that I analyzed was etcpgen. etcpgen was a software component that processed raw packets into the EMERALD message format. Mr. Porras has testified that, as of the filing date of the patent, etcpgen was SRI's best mode for performing "network surveillance" or "network monitoring" as required by the claims.[186] etcpgen had a number of useful features. For example, it could analyze a number of different network packet formats, including TCP, IP, UDP, ICMP, and ARP. Based on my review of the source code that SRI placed into escrow, it is my opinion that etcpgen was the inventors' best mode for performing network monitoring. The files listed in Exhibit DD were part of the code base for etcpgen as of November 9, 1998.

406. Having determined that etcpgen was the inventors' best mode for performing network monitoring, I then analyzed the appendix to the patents-in-suit to determine whether code for etcpgen was contained in the appendix to the patents in suit. I found none of the source code files for etcpgen—listed in Exhibit DD—in the appendix to the patents-in-suit.

407. I also analyzed the shared written description of the patents-in-suit and the articles incorporated by reference into the written description. I found no mention of etcpgen.

[186] Porras 30(b)(6) Tr. 173:10-174:24.

408. To summarize, it is my opinion that etcpgen was the inventors' best mode for performing network monitoring. I have also determined that etcpgen was not disclosed. Therefore, by failing to disclose etcpgen, the inventors failed to disclose their best mode for performing network monitoring.

E. eResolve, SRI's best mode for performing response, was not disclosed

409. Another SRI software component that I analyzed was eResolve. I understand Mr. Porras has testified[187] that either estat or eResolve was SRI's best mode for practicing "responding" or "invoking countermeasures" (generically "response") as claimed in the patents-in-suit. eResolve was a software component that received alerts from both estat and eXpert. eResolve allowed the administrator of the instrusion detection system to set a policy regarding which types of alerts to report to a higher-level entity.

410. I have determined that the files listed in Exhibit EE are part of the code base for eResolve as it existed prior to November 9. 1998. Based on my analysis of these files, I have determined that prior to November 9, 1998 SRI had implemented a complete version of eResolve.

411. None of the files listed in Exhibit EE are contained in the appendix to the patents-in-suit. Furthermore, there is no mention of eResolve in the patents-in-suit or the articles incorporated by reference into the patents-in-suit.

412. To summarize, it is my opinion that eResolve was the inventors' best mode for performing response. I have also determined that eResolve was not disclosed. Therefore, by failing to disclose eResolve, the inventors failed to disclose their best mode for performing response.

[187] Porras 30(b)(6) Tr. 180:7-183:7.

XIII. ENABLEMENT AND SUFFICIENT WRITTEN DESCRIPTION OF THE PATENTS-IN-SUIT

A. Legal standard

413. I understand that the specification of a patent must provide an enabling disclosure. I understand that this requires that a person of skill in the art, using knowledge available to them and the disclosure in the patent, could make and use the invention without undue experimentation. I also understand that the enablement standard for prior art publications is similar to that required for a patent specification.

414. I have been informed that the factors to be assessed in determining whether experimentation is "undue" include: the quantity of experimentation necessary, the amount of direction or guidance presented, the presence or absence of working examples, the nature of the invention, the state of the prior art, the relative skill of those in the art, the predictability or unpredictability of the art, and the breadth of the claims.

415. I also understand that the specification of a patent must describe the subject matter claimed in the patent in a manner that conveys to one of skill in the art that the inventors had possession of the subject matter claimed at the time the patent application was filed.

B. Analysis of enablement / written description

416. I understand that SRI contends that the disclosures in certain prior art references, including *Emerald 1997*, are not enabling for certain claim limitations.[188] It is my opinion that the prior art references discussed in my report, including *Emerald 1997*, are enabling and provide a disclosure at the same general level of detail as found in the

[188] For example, SRI has claimed that Emerald 1997 does not provide an enabling disclosure of a statistical detection method. *See* SRI International, Inc,'s "Amended" Response to Symantec's Invalidity and Inequitable Conduct Contentions (Dec. 16, 2005).

specification of the patents-in-suit. In particular, given the overall similarity between the disclosures in *Emerald 1997* and the patents-in-suit, including substantial portions of identical text and identical figures, it is not plausible to claim that one is enabled, but the other is not.

417. In particular, with regard to the disclosure in the patents-in-suit regarding statistical profiling / statistical detection method, I believe the written description of the patents alone sufficiently enables statistical profiling. As noted previously in my report, SRI has stated that the *Statistical Methods* paper is not "essential material" as defined by the USPTO, which means the paper is not required in order for the patents to be enabled. In my opinion, the algorithms disclosed in *Statistical Methods* would not be required for one of skill in the art to use the disclosures in the patents-in-suit to perform statistical profiling generally.[189]

418. In addition, I do not believe that the code included in the appendix to the patents-in-suit is required in order for the patents to be enabled. One of the inventors, Mr. Valdes, agreed.[190] The 1000+ pages of code in the appendix do include large portions of SRI's estat code base, which was used to implement statistical profiling. However, this code provides minimal commentary, and thus is extremely difficult to understand. I do not believe it would have been practical for someone with no familiarity with the code to reverse-engineer the statistical profiling algorithms from the appendix.[191] It would have been simpler for one of skill to use the patent specification's description of

[189] However, even if SRI contents the algorithms in *Statistical Methods* are required for enablement, this paper was publicly available as of 1995 and thus these algorithms were already known in the field. These algorithms would have been obvious to combine with a system such as that disclosed in *Emerald 1997*.
[190] Valdes Tr. 561.
[191] Valdes Tr. 558-559 (stating that it would be "much harder" to reverse engineer the statistical profiling algorithms from the source code than the NIDES algorithms).

statistical profiling combined with his or her existing knowledge of intrusion detection to implement a statistical profiling method.

419. To the extent that SRI claims that particular pieces of prior art, including *Emerald 1997*, are not enabling, my opinion is that this would necessitate a finding that the patents-in-suit themselves similarly do not satisfy the enablement and written description requirements.

420. I understand that certain SRI personnel, including the inventors of the patents-in-suit, have continued to file additional patent applications after the filing of the '338 application that led to the patents-in-suit. It is my opinion that the alleged inventions disclosed in these later-filed applications are not described in the patents-in-suit.

421. For example, after filing the application that matured into the '338 patent, the inventors filed applications relating to the use of Baysian algorithms in intrusion detection.[192] As the inventors have admitted, Baysian algorithms are not disclosed in the patents-in-suit and therefore are not part of the alleged inventions claimed in the patents-in-suit.[193] Similarly, the inventors also filed patent applications on later work done on alert correlation methods such as probabilistic alert correlation.[194] As discussed previously in my report, the patents-in-suit provide a very minimal description of correlation. Later work done by SRI on correlation methods such as probabilistic alert

---

[192] *See, e.g.*, A. Valdes, K. Skinner and P. Porras, Application No. 09/653,066 "Methods for Detecting and Diagnosing Abnormalities Using Real-Time Bayes Networks," filed 9/1/2000; A. Valdes, M. Fong and P. Porras, Application No. 09/952,080 "Prioritizing Bayes Network Alerts," filed 9/13/2001.
[193] Porras 30(b)(6) Tr. 250-251; Valdes Tr. 231-232, 315-316.
[194] *See, e.g.*, A. Valdes and K. Skinner, Application No. 09/944,788 "Probabilistic Alert Correlation," filed 8/31/2001.

correlation is not disclosed in the patents-in-suit and therefore is not part of the alleged inventions claimed in the patents-in-suit.[195]

XIV. PUBLIC AVAILABILITY OF CERTAIN DOCUMENTS

422. I am a named author on the paper: Steven Snapp et al., "Intrusion Detection Systems (IDS): A Survey of Existing Systems and A Proposed Distributed IDS Architecture" CSE-91-7 (February 1991) ("DIDS February 1991") [SYM_P_0069280-SYM_P_0069297]. This paper was publicly available as of 1991. The "CSE-91-7" indicates that this paper was a technical report, filed with the UC Davis Division of Computer Science. Such technical reports could be requested from the Division and were publicly available. In fact, this paper has been cited by other authors not associated with UC Davis, see, e.g., G. White et al., "Cooperating Security Managers: A Peer-Based Intrusion Detection System," IEEE Network, Jan./Feb. 1996 at [9].

423. I am a named author on the paper: B. Mukherjee et al., "Network Intrusion Detection" IEEE Network, Vol. 8 No. 3, pp. 26-41, May/June 1994 [SYM_P_0069263-SYM_P_0069279] see also [SRI 058251-058266]. As indicated by the IEEE Network magazine cover page, this article was published and publicly available as of the indicated date.

424. I have spoken to Mr. Staniford, a named author on the paper: Staniford-Chen, S., et al. "GrIDS - A graph based intrusion detection system for large networks," 19th National Information Systems Security Conference, 1996 ("GrIDS 1996") [SYM_P_0068883-SYM_P_0068892]. Based upon our discussion, I understand that this paper was published and publicly available in the 19th NISSC conference proceedings, which were distributed to all conference attendees.

---

[195] Valdes Tr. 234.

425. I have spoken to Mr. Staniford, a named author on the paper: Steven Cheung, Rick Crawford, Mark Dilger, Jeremy Frank, Jim Hoagland, Karl Levitt, Stuart Staniford-Chen, Raymond Yip, Dan Zerkle, "The Design of GRIDS: A Graph-Based Intrusion Detection System," Technical report, UC Davis Department of Computer Science, Davis California (May 14, 1997) (GrIDS 1997") [SYM_P_0080878-SYM_P_0080943]. Based upon our discussion, I understand that Mr. Staniford posted this paper to the GrIDS home page by 1996. The document was regularly updated through 1997. Furthermore, the Internet Archive demonstrates this paper was publicly available at least as early as July 19, 1997. *See* SYM_P_0512090 - SYM_P_0512181.

426. I have spoken to Mr. Smaha, the author of the Stalker line of products. Based upon our conversation, I understand that NetStalker, Installation and User's Guide, Version 1.0.2 1996 [SYM_P_0079550- SYM_P_0079629] was distributed to customers with versions of the NetStalker product prior to Nov. 1997. I also understand that employees of Tivoli actually tested the NetStalker product with Tivoli, a network management application.

427. Based upon my review of the University of New Mexico's on-line catalog, I believe the publication: Richard Heady, George Luger, Arthur Maccabe, and Mark Servilla, "The Architecture of a Network Level Intrusion Detection System," Technical Report CS90-20, University of New Mexico, Department of Computer Science, August 1990 [SYM_P_050086- SYM_P_0500603] was cataloged at UNM and publicly available prior to Nov. 9, 1997. *See* [SYM_P_0535342].

428. As indicated in the Hansen and Berard declarations in Exhibit HH, the thesis: Feather, Frank Edward, Ph.D., "Fault Detection in an Ethernet network via anomaly detectors," Carnegie Mellon University, Order number 9224199 (1992) [SYM_P_0501779- SYM_P_0502036] was cataloged and publicly available prior to Nov. 9, 1997.

432. I have spoken to Mr. Staniford, the moderator for the CIDF mailing list. Based upon my conversation with him, as well as my own personal recollection and my review of Internet Archive documentation relating to CIDF, it is my opinion that documentation and emails distributed to the CIDF mailing list were archived publicly on the CIDF website contemporaneously with their distribution. *See, e.g.*, Mr. Stanford's email setting up this archive [SYM_P_0603086].

Dated: April 21, 2006

Louis Todd Heberlein

L. Todd Heberlein